"Express Mail" Mailing Label No. EL 521 508 534 US
Date of Deposit October 27, 2000

I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above, with sufficient postage affixed, and is addressed to BOX PATENT APPLICATION, Assistant Commissioner for Patents, Washington, D.C. 20231.

Lucy Flemings

(Typed or printed name of person mailing paper or fee)

L. Flemings

(Signature of person mailing paper or fee)

PATENT
Atty Dkt. No. 032001-051

METHOD AND IMPLEMENTATION OF A TRACEBACK-FREE PARALLEL VITERBI DECODER

BACKGROUND OF THE PRESENT INVENTION

The present invention relates to implementations of the Viterbi algorithm, especially implementations that parallelize some of the steps of the Viterbi algorithm.

Viterbi decoding was developed by Andrew J. Viterbi. The seminal paper on the technique is "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm" published in *IEEE Transactions on Information Theory*, Vol. IT-13, pages 260-269, in April 1967. Viterbi decoding has been found to be optimal when the channel of the transmitted signal is corrupted by additive white Gaussian noise (AWGN). AWGN is noise whose voltage distribution over time has a characteristic that can be described using a Gaussian distribution (normal statistical distribution or bell curve distribution).

The Viterbi algorithm uses a trellis which restricts the number of state transitions. In the Viterbi algorithm, each new state has a specified number of possible state transitions from previous states. A branch metric comparing a selected value to an ideal value for a transition is calculated for each transition. The branch metric value is combined with a prior state path metric value, in order to produce updated candidate path metrics. For each new state, the candidate path metric with the lowest value is selected. An indication of the selected transition into the new state for that symbol is also stored.

Fig. 1 shows a conventional block diagram of a conventional Viterbi algorithm. In block 22, a branch metric calculation is done. The branch metric calculation compares the actual received value with the ideal received value for different symbol transmissions. In a binary symbol encoding system, for example, one of the transitions going into a state corresponds to the transmitted symbol "0," and one of the transitions corresponds to the transmitted symbol "1." The difference between the input symbol value and the value that would be received in the ideal case if a "1" is transmitted is the branch metric for the "1" transition and the difference between the received sample value and the ideal value if a "0" is transmitted would be the branch metric for the other branch. The branch metrics are added to the old value of the path metric for the source states. The two new candidate path metrics are compared to select the smallest (lower-energy) path metric. Unit 24 is typically called an add-compare-select (ACS) unit. The updated path metrics are stored in a path metric memory 26. In the system of Fig. 1, when the new path metric is produced, a traceback pointer is stored in the traceback pointer memory 28. The traceback pointer indicates the transition into the new state for one symbol. Trackback pointers are stored for every state of every symbol.

In the simplest state, the convolution encoder is reinitialized to an all-zero state at the end of a transmitted block of data. This typically means that at the receiver, state "0" is assumed to be the correct state after the transmission of the

block. In the traceback algorithm, the data for each symbol transmitted needs to be examined to determine the transition of the decided path. This typically is a serial operation which takes at least as many steps as symbols within the transmitted block.

It is desired to have an improved method which speeds up the operation of the Viterbi algorithm.

SUMMARY OF THE INVENTION

One embodiment of the present invention comprises a Viterbi algorithm using an optimal path value generator for each state in the trellis, the optimal path value indicating more than one transition of the selected trellis path. This optimal path value can then be used to determine the output in fewer steps than the conventional traceback. In a preferred embodiment, the old optimal path for the source state transitioning into the new state is appended with new data indicating the selected state to produce the new optimal path value. The first selected optimal path value will indicate the best estimate of the transmitted symbols.

One embodiment of the present invention comprises a method of implementing the Viterbi algorithm comprising calculating branch metrics for branches of the Viterbi trellis, combining branch metrics with old path metrics to produce candidate path metrics, selecting a new path metric associated with a selected trellis path for each state in the trellis from the candidate path metrics, and composing an optimal path value for each state in trellis, the optimal path value indicating multiple transitions of the selected trellis path.

Another embodiment of the present invention is an apparatus to implement the Viterbi algorithm comprising a path metric storage adapted to store a path metric associated with a selected trellis path for each state in the Viterbi trellis, a path update unit adapted to update each path metric, an optimal path value storage adapted to store an optimal path value for each state in the trellis, the optimal path

value indicating multiple transitions of the selected trellis path, and an optimal path value update unit adapted to update each optimal path value.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a diagram of a conventional Viterbi algorithm system.

Fig. 2 is a diagram illustrating a system which constructs optimal path values for each state in the Viterbi trellis.

Fig. 3 is a diagram of one example of an optimal path update of one embodiment of the present invention.

Fig. 4 is a diagram of an example of a 16-state Viterbi trellis.

Fig. 5 is a diagram of the calculations which are done for each state and each symbol in the system of one embodiment of the present invention.

Figs. 6A - 6C are diagrams that illustrate the time improvement which can be obtained with the system of the present invention.

Fig. 7 is a diagram that illustrates one embodiment of an implementation of the optimum path memory in one embodiment of the present invention.

Fig. 8 is a flow chart that illustrates the parallel operation of the new path metric calculation and new optimal path calculations in one embodiment of the present invention.

Fig. 9 is a diagram that illustrates a reconfigurable chip system which can be used to advantageously implement the parallel Viterbi algorithm of the present invention.

Fig. 10 is a diagram that illustrates an example of a 1/4 convolution encoder.

Fig. 11 is a diagram of an example of an optimal path update unit for one implementation of the present invention.

Fig. 12 is a diagram of a branch metric calculation circuit in one embodiment of the present invention.

Fig. 13 is a diagram of an add/compare/subtract circuit used with one embodiment of the present invention;

Fig. 14 is a diagram of a path metric update circuit of one embodiment of the present invention.

Fig. 15 is a diagram of an optimal path value construction unit of one embodiment of the present invention.

Figs. 16A and 16B are diagrams illustrating a "ping-pong" memory embodiment of Fig. 6C.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

Fig. 2 is an example of a block diagram of the parallel Viterbi algorithm 40 used in the present invention. The branch metric calculation block 22', path metric update 24' and path metrical store 26' operates as in the conventional Viterbi unit.

The optimal path update block 42 produces optimal path data for each state in the trellis. The optimal path value indicates transitions in the minimum-energy path from the beginning of the block to that trellis state. To update the optimal path, optimal paths for the source states transitioning into the current trellis state are loaded into the optimal path update 42 from the optimal path memory 44. New data indicating the selected transition into the trellis state is appended to the end of the optimal path value of the selected prior state. Eventually, after all the blocks of the symbols are transmitted, the optimal path memory 44 for the lowest energy state will indicate the optimal path through the trellis for all the symbols and thus be able to be used to determine the output symbols. In a preferred embodiment, the "new data" appended for the optimal path value indicates the transmitted symbol.

Note that the traceback operation is significantly sped up. In one embodiment, if the optimal path value is stored in the memory without being fragmented, the read-back could take a single step. As will be described below

with respect to Fig. 7, in a preferred embodiment, the optimal path value will be fragmented and placed into multiple memory blocks to simplify processing; however, the operation of the traceback is still significantly sped up.

Fig. 3 illustrates the operation of an optimal path update unit 50 of one embodiment of the present invention. Previous optimal path values of the source states into the current trellis state are loaded from the optimal path memory 52. As will be described below, this loading of the previous optimal paths can be done in a preloading step. Multiplexer 54 is used to select the desired previous optimal path using the traceback pointer value produced by the path metric update unit. The traceback pointer is an indication of which of the transitions into a state is selected in the ACS operations of the path metric update unit. The optimal path constructor unit 56 appends "new data" to the old optimal path. In a preferred embodiment the new data is shifted into the least significant bits. The new data determinator 58 determines the "new data." In one embodiment the new data is data indicative of the estimated value of the transmitted symbol. This also gives an indication of the preferred transitions in the selected trellis path. The number of previous optimal paths selected from is determined by the number of transition paths going into a single state in the trellis. For binary transmission systems, two transitions are used, and a single bit of new data is added to the optimal path value. For quatranary transmission systems, four previous optimal paths (prior states) are selected from, and two bits of new data appended to the optimal path value.

Block 60 indicates a prior state determination used to produce the prior state addresses into the optimal path memory. Note that the prior states for the optimal path update are the prior states which are needed for the path metric update in the path metric update portion of the Viterbi algorithm. Thus, the addressing systems of the two portions can be shared.

Fig. 4 illustrates a 16-state convolution encoder state transition graph for a rate one-half convolution code. Fig. 1 illustrates an example of a transition

calculation for one embodiment of the present invention. Note that the new state So can have a previous state So or S8. In the Viterbi algorithm of the present invention, a branch metric is calculated for each transition between the states. The branch metric is calculated for each of the possible transitions between states in the trellis. In this example, the branch between old state So and new state So has a branch metric of 1. The branch between the old state S8 and new state So has a branch metric of 3. After the calculation of the branch metric, the branch metric is added to the prior path metric of the old state to produce two candidate path metrics. One of the candidate path metrics corresponding to the transition between state So and state So has the path metric of 6; the other candidate path metric corresponding to the transition between old state S8 and new state So has a candidate path metric of 16. Since the path metric of 6 is less than candidate path metric 16, the new path metric is selected as 6, and the selected transition is a transition between old state So and new state So.

Looking again at Fig. 5, the old state S_0 had a prior optimal path of "...1011," whereas the old state S_8 has a prior optimal path of "...1100." The optimal path associate with old state as "0" if shifted one bit to allow the new data, "0," indicative of the transition between the state S_0 and s0 to be added to the produce a new optimal path "...10110."

Fig. 5 makes it clear that updating the optimal path values is computationally intensive. Thus, the method of the present invention doesn't make much sense on a single processor system. However, if the optimal path update can be parallelized with the new path state update, the improvement of the serial traceback speed at the end will improve the total of the entire algorithm. This is illustrated in Figs. 6A and 6B.

Fig. 6A illustrates the conventional algorithm in which the serial traceback 72 is done after the path metric calculations. In the system of the present invention, while the path metric calculations 70' are done, the optimal path calculations 74 are done in parallel. Even though these optimal path calculations

74 are extensive — in fact, much more extensive than the traceback calculations 72 shown in Fig. 6A — the total time is reduced.

Fig. 6C shows an embodiment in which the readout steps are done in parallel with the processing steps for the next block of symbols. This further speeds up the operation of the Viterbi algorithm. The readout operation can operate on a previous optimal path value memory while the current optimal path value memory is accessed by the optimal path value update operation.

Figs. 16A and 16B show a "ping-pong" memory that can be used with the system of Fig. 6C. In Fig. 16A, memory 140 is used in the optimal path value update for symbol block A. Looking at Fig. 16B, after the optimal path value for symbool block B is completely loaded into memory 140', the function of memories 140' and 142' flip. Memory 140' is used for the readback and memory 142' is used for optimal path value update.

Fig. 7 illustrates one embodiment of the present invention wherein the optimal path memory is broken into a number of smaller memory blocks. The breaking of the memory into a number of smaller memory blocks allows the processing to be done on smaller sized optimal path value fragments, rather than the entire optimal path value. In this embodiment, the updated optimal path fragments are written into a memory block. For example, in one embodiment, the memory block 82 is written into first. When the memory block 82 is filled with data, the next memory block 84 is written into. This is done until memory block 86 is written into, such that all optimal path data for the entire block of transmitted symbols is stored. The current block pointer 90 tells which of the memory blocks to write the updated path value fragments into, and from which memory block to obtain the optimal path fragments of the prior states. Note that in a preferred embodiment, the optimal path value fragment stored in each of the filled memory blocks is not later modified. In this preferred embodiment, the system produces a pointer to the address in the previous memory block of the next fragment of the optimal path for a state. Once a memory block is filled, that fragment of the

optimal path will remain at the same address corresponding to the state of the trellis when the block is filled.

In one embodiment, the address pointer can be directly determined from bits within the previous optimal path value fragment. Looking at Fig. 10, the information bits are read into the convolution encoder which is implemented as a shift register. The optimal path value for the selected lowest energy state will, in a preferred embodiment, be the transmitted symbols of data. Consider the situation where the transmitter and the convolution encoder have transmitted symbols corresponding to a given state. When a memory block is filled up, the address of the optimal path fragment stored in the memory block will be at the address corresponding to the state of the convolution encoder when the block is filled up. The next bits transmitted from the convolution encoder are then decoded and placed in the optimal path value fragment stored in the next memory block. These next bits will also give the address of the previous optimal path fragment in the previous memory block.

When a full symbol block is transmitted, the optimal path corresponding to the state with the lowest path metric is selected. In some embodiments, the block of data is stuffed with zeros to cause the final state of the transmitted block of symbolsto be state zero. The optimal path value fragment from this state is read out from the memory block 86. The most significant bits of this optimal path fragment will indicate an address of the fragment in the memory block 85. This optimal path value fragment is then read out and the most significant bits point to the optimal path value fragment in the prior memory blocks, and so on. This is done until all of the optimal path value is read out of the memory blocks.

By breaking the optimal path value into optimal path fragments, the memory stores and optimal path value operations shown in the previous figures can be more efficiently done. In this embodiment, a number of optimal path value fragment reads equal to the number of memory blocks are done. In one embodiment in which there are 192 symbols in a block and each memory block is

set up to be 32 bits wide, six memory blocks are used to implement the optimal path memory.

In an embodiment in which the most significant bits in the optimal path fragment do not effectively act as a pointer to the address in the prior memory block, bits can be stuffed into the memory blocks in order to act as such a pointer. The downside of this embodiment is that it may increase the number of memory blocks required.

Fig. 8 illustrates a flow chart illustrating the operation of the parallel Viterbi method of the present invention. In the path metric calculations 98, the branch metrics for a state transition are calculated in step 100. In a preferred embodiment in which the convolution encoder is zeroed out at the end of a block, the system knows that the previous state is state zero. In step 102, the candidate path metrics are calculated by adding the calculated branch metrics to previous path metric data. In step 104, a new path metric is selected. This new path metric can be then stored in the path metric storage. Step 106 determines whether every state for a symbol has had its path metric updated. If not, the system goes, in step 108, to the next state in the trellis. After every state for a symbol is checked, in step 110 it is checked whether every symbol in the block has been operated on. If not, the next symbol is moved on to in step 112.

The optimal path calculation 114 preferrably operates in parallel with the calculation of the new path metric. In step 116, the construction of the path metric produces an indication of the traceback pointer. This traceback pointer allows for the determination of the new data in step 116, and the updating of the optimal path in step 118. The steps 120 effectively duplicate the steps 105. For this reason, indications from the calculation of the new path metrics can be used to update the next data and the next symbol. In one embodiment, in order to speed up the operation of the steps 116 and 118, the previous optimal paths are preloaded in a step 122. The previous optimal path values can be then updated in the updating step 118. Note that the calculation of the optimal path can be quite

computationally intense, requiring calculations for every trellis state in each symbol period. Thus, if there are 256 trellis states and 192 symbols in the block, the number of updates of the optimal path in the calculation steps 114 is 256 x 192. Due to the parallelism, calculation 114 steps are in fact done at the same time as new path metric block calculation steps 198. Thus, the readout of the optimal path in step 124 can be made much quicker than the traceback technique done in the prior art, reducing the total calculation time.

Fig. 9 illustrates a reconfigurable chip. In the reconfigurable chip, background and foregoing planes of configurations, such as the configurations used in the Viterbi algorithm, can be loaded to configure the reconfigurable fabric. In this system, the resources of the reconfigurable fabric are typically fixed. Thus there can often be a situation in which the configuration used to calculate the path metrics leaves enough fabric resources to allow the implementation of the optimal path updating logic. This means that the optimal path updating logic can be implemented without requiring the use of an additional reconfigurable chip. Thus the time savings in using the optimal path value construction can be produced without a resource penalty. Note that this might not always be the case.

Fig. 10 illustrates an example of a convolution encoder in a transmitter. The bits in the linear shift register correspond to the state of the Viterbi trellis. The system shown in Fig. 10 is a one-fourth convolution encoder in which four output bits are provided for each input bit. As each new information bit is put into the convolution encoder of Fig. 10, the linear shift register of the convolution encoder can go into only two new possible states. This corresponds to the two possible transitions in the Viterbi trellis. Each transition is associated with a new input information bit, which causes the state transition. These two different states of the convolution encoder cause the four output signals, C0, C1, C2 and C3, to be one of two different patterns corresponding to the two different possible states. At the detector, the four signals C0-C3 are transmitted and received. Estimates of the value for C0-C3 are then produced. Since the convolution encoder can be in

only two states, the detector can calculate the ideal values for C0-C3 for both of the two states. The deviation of this ideal value of C0-C3 from the real values of C0-C3 are used to produce the error signal or branch metric for each of the transitions in the trellis, as is shown in more detail in Fig. 12.

In a preferred embodiment, each block of symbols ends in enough states to clear the convolution encoder into state "0." This means that the final state of a block is state "0," and the optimal path value for state "0" can be selected at the end of the block. Other Viterbi implementations require the use of the lowest total path metric to select the correct optimum path value.

Fig. 11 illustrates the optimal path memory update algorithm block diagram of one embodiment of the present invention. Looking again at the convolution encode example of Fig. 10, note that if there are k states in the convolution encoder and the convolution encoder is in state m, the next possible states are $(2m) \bmod 2^k$ or $(2m+1) \bmod 2^k$. From the current state n it is possible to determine the previous states in a similar manner. The previous state p is equal to the values of bits in registers S1 - S7 shifted into locations S0 - S6 with a zero or a one in the S0 position. This corresponds to the previous memory state: p is equal to the floor function of $n/2$ or the floor function of $n/2 + 2^{k-1}$. The addressing for the previous optimal path selection and for the previous path metrics is determined in this fashion.

Looking again at Fig. 11, the optimal path value or optimal path value fragments are sent to a multiplexer. The traceback pointer produced by the add/compare/select unit is used to send the selected optimal path value or optimal path value fragment into a unit that adds the new bit into the least significant bit and shifts the other bits. This is now used as the new optimal path value for the memory state n.

Fig. 12 illustrates the branch metric calculation circuit. Note that for binary phase shift keying (BPSK), a "1" is transmitted as a positive one and a "0" is transmitted as a negative one. In the detector, a 16-bit detected value for each

transmitted code C0-C3 is created. In the ideal case these detected values would be either one or negative one but, due to noise and other effects, there can be a significant variation in the detected signal strengths. The deviation of four detected signals X0-X3 from the ideal values of the transmitted C0-C3 signals for a transition in the trellis is used to determine the branch metric. Details of one embodiment of the branch metric calculation are given in the Appendix.

Fig. 13 illustrates a general add/compare/select circuit used in one embodiment of the present invention.

Fig. 14 illustrates another path metric state update circuit.

Fig. 15 illustrates an implementation of a single optimal path value update circuit.

Details of the implementation of the parallel Viterbi algorithm on a reconfigurable chip is given in the Appendix entitled Design and Implementation of a Parallel Viterbi Decoder.

It will be appreciated by those of ordinary skill in the art that the invention can be implemented in other specific forms without departing from the spirit or character thereof. The presently disclosed embodiments are therefore considered in all respects to be illustrative and not restrictive. The scope of the invention is illustrated by the appended claims rather than the foregoing description, and all changes that come within the meaning and range of equivalents thereof are intended to be embraced herein.